**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 8, No. 2, June 2019, pp. 132~143
ISSN: 2252-8938, DOI: 10.11591/ijai.v8.i2.pp132-143


# Suggestive GAN for supporting Dysgraphic drawing skills

**Smita Pallavi[1], Akash Kumar[2], Abhinav Ankur[3]**
[1,2]Department of CSE, Birla Institute of Technology Mesra Patna Campus, India
[3]Data Science Analyst, Verizon India, Hyderbad, India

**Article Info**

***Article history:***

Received Feb 24, 2019
Revised Apr 15, 2019
Accepted May 9, 2019

***Keywords:***

Autoencoder
Dysgraphia
Generative Adversarial Networks
infoGAN
Suggestive GAN

**ABSTRACT**

The squat competence of dysgraphia affected students in drawing graphics on paper may deter the normal pace of learning skills of children. Convolutional neural network may tend to extract and stabilize the action-motion disorder by reconstructing features and inferences on natural drawings. The work in this context is to devise a scalable Generative Adversarial Network system that allows training and compilation of image generation using real time generated images and Google QuickDraw dataset to use quick and accurate modalities to provide feedback to empower the guiding software as an apt substitute for human tutor. The training loss accuracy of both discriminator and generator networks is also compared for the SGAN optimizer.



***Corresponding Author:***

Smita Pallavi,
Department of Computer Science Engineering,
Birla Institute of Technology, Mesra,
Patna Campus, Opp. B.V. College, Patna-800014, Bihar, India.
Email: smita.pallavi@bitmesra.ac.in

## 1. INTRODUCTION

Reconstruction of complete original images from partial images involves identification of probable semantic features that the image might exhibit to keep the right configuration of the object. The goal of such processes is identification of the unknown original image and getting close to the orientation of the image. The problem intended in this work is to solve the partial occlusion problem which limits the recreation of image information. Researchers have identified that stereo vision occlusion occurs when a portion of the picture visible on one image is occluded in the other by the scene itself or, a section of the scene near the image boundary moves out of the field of persuasion on the other picture. We correlated the problem of partial occlusion among special children and have devised a model GAN algorithm to assist in semantic image synthesis.

Dysgraphia is a neurological syndrome affecting the development of the brain consequently hindering the fine motor skills. Human brain retains information based on visions and writings/drawings performed at early stages of learning. When such processes are hampered biologically, the natural development goes through major setback deteriorating the growth and development at adolescence. Diagnosing Dysgraphia is still a major challenge due to its enigmatic nature. Parents confound these disorders with the puerile behaviour of children. We propose an adversarial network architecture suggestive solution to assist a dysgraphic human in gaining control over their representational skills.

The network assimilates initials of a dysgraphic sketch and reconstructs the probable features according to the models trained. This acts as a guiding framework for the muscular activity of the convalescent. The work emphasises that application of Autoencoder and suggestive GAN on human-made sketch dataset yields a realistic guiding model with prodigious accuracy. GAN and Autoencoders were first combined by Mescheder et.al [1]. The improvements exhibited by SGAN are based on additionally learning

the semi-constructed drawing features that are fed into the classifiers. The motivation to involve deep learning algorithms for a societal work is because these algorithms have shown extremely high performance on machine learning tasks such as image recognition and classification.

A.Makhzani et.al [2] devised Adversarial Autoencoder (AAE) that performed variational inference by matching the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior distribution. There are various more works on combined Autoencoders and GANs [3-5]. The remainder of this paper is organized as follows: Section 2 mentions about the related literature review and its shortcomings. Section 3 describes about the methods involved in implementing the various deep learning optimizers. Section 4 discusses the proposed SGAN pseudocode and the dataset used for training and testing data methods. Section 5 discusses about the SGAN experimentation and associated results. Finally, we conclude with summary and further research.

## 2. RELATED LITERATURE REVIEW

In recent past, the related work with ability to predict image protypes with GAN studying pattern can be listed as follows. Mescheder et.al [1] first combined Autoencoder and Generative Adversarial Networks. They coined Adversarial Variational Bayes as a tool for constructing variational autoencoders with equivalent inference models which were arbitrarily expressive (AVB). They established principled connection between VAEs and GANs by implementing an auxiliary discriminative network which suggests a two playergame in congruence with the maximum-likelihood-problem. S.Pallavi et.al have used Generative Adversarial Networks for image generation using MNIST dataset. They have compared the training loss accuracy of AdaGrad optimizer withthat of the Adam optimizer. Xiaoguang Han, Chang Gao, and Yizhou Yu. [6] researched on reconstruction of 3-dimensional model features using simple human sketch. The model was however limited to human face prototype. Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh [7] devised a system which considered view rotation, axis selection and modified sketching techniques to render the 3-dimensional models of sketches. Yang Song et.al [8] worked on a recursive cross-domain face/sketch generation which took into account the availability of partial images. This model synthesized sketches with 90% data. Our model shows improvement by working on just 25% of the original sketch. Further 3D modelling just by using few strokes was experimented by Cherlin, J.J. et.al [9].

## 3. DEEP LEARNING METHODS

Learning Analytics has grown manifold with dedicated software and faster generation of tools with domain-specific libraries and programming packages as Python, R and Weka. GANs are neural networks with capabilities to generate synthetic data with certain input data provided to the network. GANs have been taught in the previous literature work to generate images from text. Generative Adversarial Network is the unsupervisedmachine learning AI tool introduced by Ian Goodfellow et al. [10]. The combinatorial framework consists of two nets, agenerative model further pitted to an adversary discriminative model. While the generative model generates new data instances, evaluation of these generated data instances is carried out by the discriminative modelfor authenticity. The implementation of multilayer perceptronsin both the models makes it convenient toimplement the adversarial modelling framework. At generator end, a mapping to data space is done by the differential generator function $G(y; \theta_X)$ where y is the noise variable and parameter $\theta_X$distributes over data X. The discriminator D gives a scalar function $D(z; \theta_y)$where z is the data element not belonging to the generator's distribution. both D(z) and G(y) are trained simultaneously with G attempting to minimize $\log(1-D(G(y)))$ and D maximizes theprobability to assign the correct label to both training examples.

This optimizer's cost-value function C(G,D) can be hence implicated with D and G representing the two-player minimax problem.[10] The Generative Adversarial Network model as shown in the Figure 1 illustrates the coupled Generator and Discriminator network counteracting the former's dominance of its role. The discriminative model in a GAN operates like a normal binary classifier with ability to classify images into different classes. It is the determinant between artificially generated or real images.The generative model attempts to predict features when the classes are supplied as input. This involves determining the probability of a feature given a class.

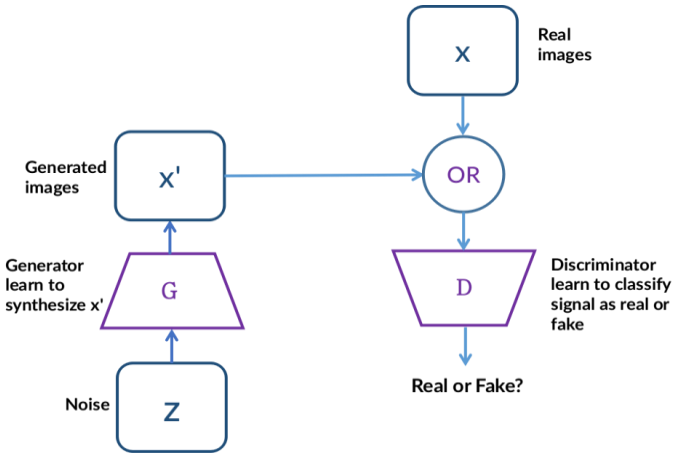



Figure 1. Generative Adversarial Networks

Generator takes noise as input and acts as counterfeit to dataset by synthesizing similar samples. Discriminator is stimulated to detect imposture. Both the adversaries are in constant battle throughout the training process. The proficiency of both the networks are mutual and dependent as shown in (1).

$$min_G max_D V(D,G) = E_{x\sim p_{data(x)}}[logD(x;\theta_d)] + E_{z\sim p_z(z)}[\log(1 - D(G(z;\theta_g);\theta_d))] \quad (1)$$

where $E_{x\sim p_{data(x)}}[logD(x)]$ in (1) signifies log likelihood of discriminator output when input are sampled from original data distribution.

$E_{x\sim p_z(x)}[log(1 - D(G(z))]$ corresponds to the log likelihood of one minus the discriminator output when the input are sampled from generated images. In the training process the optimality and equilibrium condition is the (2).

$$D(x,\theta_d) = 1/2 \quad (2)$$

when the above condition is achieved, discriminator ceases to discern between samples from dataset and generators output.

### 3.1. Autoencoders

Autoencoder incorporates a bottleneck in a deep network reconstructing input images at the output layer as shown in Figure 2. A constraint at the number of nodes in the hidden layeris enforced which facilitates lossy compression of the images [11-12]. Autoencoders are considered the most efficient lossy compressor for images. Undercomplete autoencoder sieves most relevant features by featuring non-linear characteristics unlike PCA during dimensionality reduction in a dataset [13].

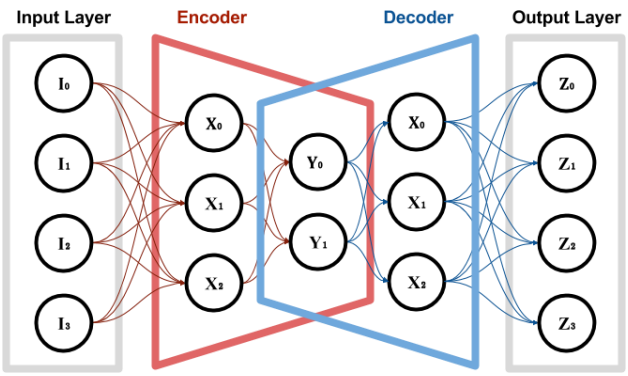



Figure 2. The Deep Network Undercomplete Autoencoder

The Autoencoder with restrained hidden layer nodes are called Undercomplete Autoencoders. The learning process aims at minimizing the loss function as in (3).

$$L(x, g(f(x))) = |x - g(f(x))|^2 \quad (3)$$

The loss function L puts a penalty on g(f(x)) for characterizing dissimilarity from x measured through mean square error as shown in Figure 2.

**a. The infoGAN**

The infoGAN allows intervention to the generator input to foster customization and control over the synthesized output as shown in Figure 3. The extension from GAN can be represented just by integrating the a simple regularization term in the GAN objective function [14].

$$min_G max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c)) \quad (4)$$

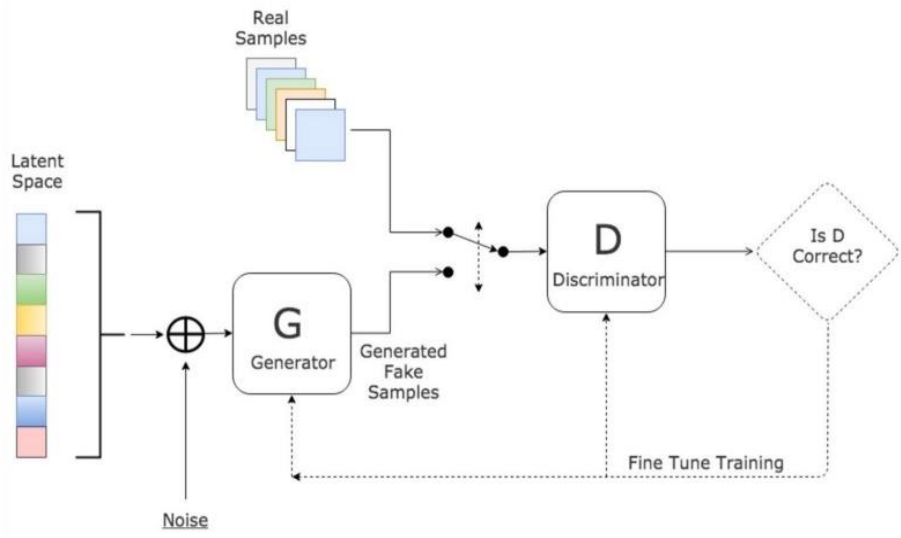


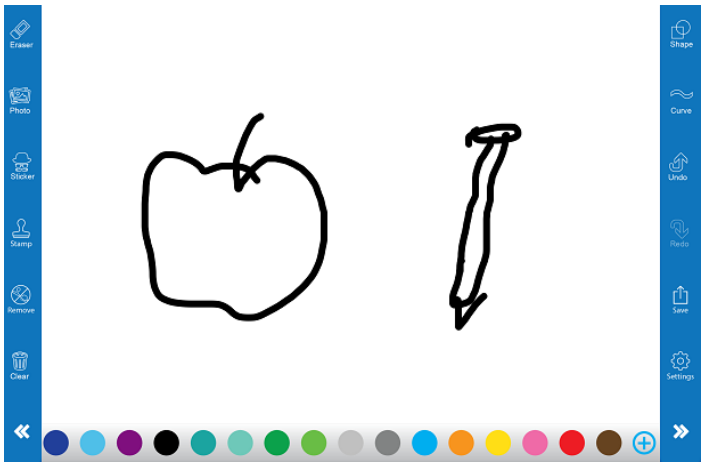

Figure 3. Introduced latent space alongsize noise in Generator input

Where I(c;G(z,c)) is the mutual information between latent code c and generated output G(z,c). Using standard variational arguments a lower bound is approximated to overcome the infeasibility of calculating mutual information [15]. An auxiliary distribution Q(c|x) is introduced which aims at approximating P(c|x) i.e. the likelihood of code c given the generated input x. The objective function is transformed to attain the form given in (5) after lower bound approximation to the mutual information.

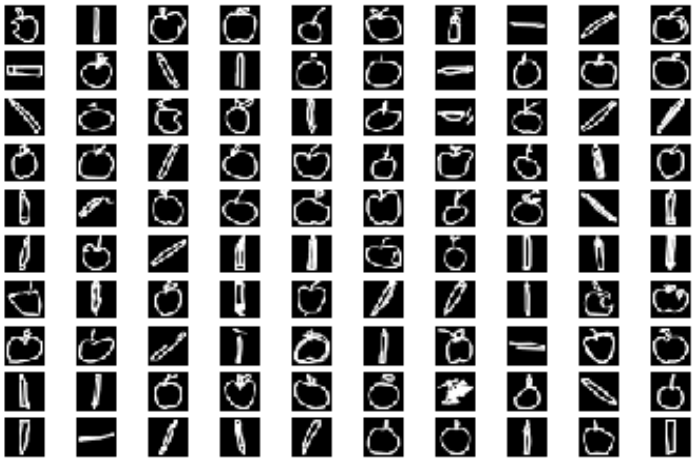
$$min_{G,Q} max_D V_{infoGAN}(D, G, Q) = V(D, G) - \lambda L_1(G, Q) \quad (5)$$

## 4. DATASET AND PROPOSED MODEL

To generate true data samples which are obtained in similarity with the feature space, the besta way could be to annotate the data pixels of the training image samples. But since this method is highly cumbersome and expensive so alternatively, a semi supervised learning approach was adopted which is explained briefly in this section. Initially, samples of a trainee suffering from dysgraphia was taken to perceive the object understanding and creation. Later the standard Google Draw dataset was used with the features drawing_ID, category(what quick draw asked the user to draw), timestamp, whether AI guessed it correct or not, user's country and drawing. The experimental data thus included dataset of 144722 apple sketches and 122001 pencil sketches i.e. a total 266723 images of 28x28 pixels as shown in Figure 4. The drawings are represented as a list of list which is basically a list of strokes and each stroke is a list of X, Y and time.




(a)



(b)

Figure 4. Sample hand drawing and dataset; (a) Sample hand drawings of the test objects taken by a dysgraphia suffering adult, (b) The sample of dataset consisting of hand-drawn apple and pencil images of size 28x28 px

The suggestive GAN takes single stroke as input defined over a quarter of the image and suggests the complete image seeking high mutual information between the single stroke and suggested image. The structured semantic features of thestrokes are derived using Undercomplete Autoencoder.

$$\phi, \psi = arg_{\phi,\psi} min|\chi - (\phi \circ \psi)\chi|^2 \text{ with } \phi: \chi \to \text{F}, \psi: \text{F} \to \chi$$
$$AE(x) = \phi(\Delta(x)) \text{ where } \Delta(\text{x}) \Longrightarrow \text{selecting first quarter of x} \quad (6)$$

Next, We need to devise new objective which keeps I(AE(x); G(z, AE(x))) high i.e. we try to achieve a state which keeps some significant amount of mutual information between the parameters fed and the output of the generator. Mutual information has been used similarly before to achieve tasks of clustering [16-17]. Thus we get information-regularized objective function as summated in (7).

$$min_G max_D V_I(D, G) = E_{x \sim p_{data(x)}}[logD(x; \theta_d)] + E_{Z \sim p_z(z)}[log(1 - D(G(z; AE(x); \theta_g); \theta_d)] - \alpha I(AE(x);\ G(z, AE(x))) \quad (7)$$

where α is the regularization rate. Through experiments the optimal learning is achieved at =0.5. AE(x) produces latent code differentiating semantics of quarter images. The information content of the AE(x) retains structures of complete image and is derived by the two sided parameter coordination i.e. autoencoder and generator.

### 4.1. The SuggestiveGAN Algorithm

We propose a modified Generative Algorithm which works upon batches of the sample image and trains them by autoencoding followed by discriminator and generator combination to reconstruct original semantics of the image as shown in Figure 5.

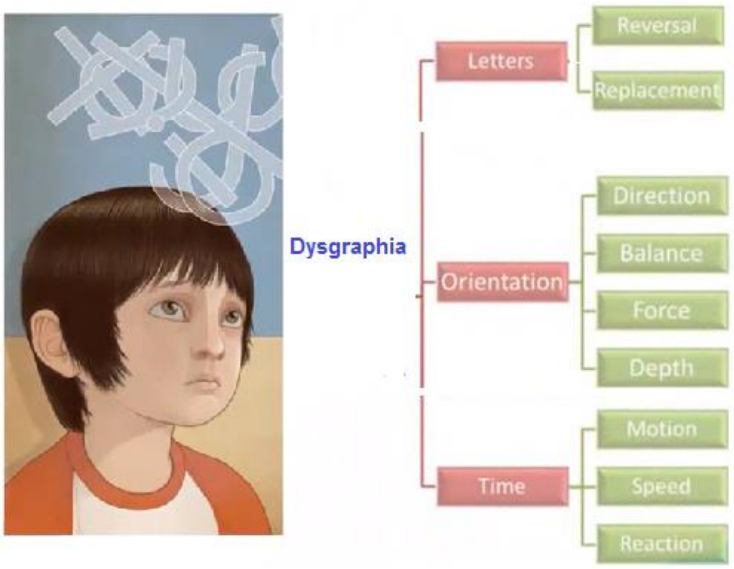



Figure 5. Theusefulness of proposed SGAN to address the problem areas of Dysgraphic sufferer

Algo SGAN (I/P: image)

| | |
|---|---|
| Discriminator | := CNN(input = *image_size*, output = class_labels \| fake) |
| Generator | := CNN(input = (noise + *dimention*_AE()), output = *image_size*) |
| Autoencoder | := DNN(input = *image_size*/4, hidden layer = *dimention*_AE(), output = *image_size*/4) |

for number of epochs do:

- batch of n samples $\{z^1, \ldots z^n\}$ where $z \sim p_z$.
- batch of n samples $\{x^1, \ldots x^n\}$ where $x \sim p_{data}$.
- train autoencoder AE using x where $x \sim p_{data}$.
- batch of n samples from function AE(x) where $x \sim p_{data}$.
- discriminator updation by the formulae

$$\nabla_{\phi_d} \frac{1}{n} \sum_{i=1}^{n} [\log D(x^i) + \log\left(1 - D\left(G\left(z^i, AE(x^i)\right)\right)\right)]$$

- generator updation by

$$\nabla_{\phi_d} \frac{1}{n} \sum_{i=1}^{n} [\log\left(1 - D\left(G\left(z^i, AE(x^i)\right)\right)\right) + \alpha I(AE(x^i); G\left(z^i, AE(x^i)\right))]$$

end for

## 5. EXPERIMENTAL RESULTS

The first step devised was to form a Deep Network Undercomplete Autoencoder. The autoencoder uses two dense layers with ReLU and sigmoid activation function. The capped hidden layer enables 83.67% compression of images retaining enough semantics to reconstruct the basic structure of the image [18].

The first quarter(x) is taken as input to the autoencoder which is passed through dense hidden layer, nodes restricted to 32. The activation function used is Rectified Linear Unit. A dense layer again reconstructs this encoded image hidden layer to the origin quarter image feeded in the network(x'). Sigmoid activation function is used in the reconstruction output layer. The Autoencoder network illustrated in Figures 6 and Figure 7 suggest that it is a neural network that simply copies the input to the output with the control to efficiently represent the data with the Encoder acting as a recognition network which converts the image input of 14x14 px into the internal representations of rectified linear units. The Decoder acts as a generative nework, converting ReLU applied by the sigmoid function internally into the output of 14x14 pixels. Thus, with the same number of neurons, the model is a reconstructive MLP (Multi-Layer Perceptron). The loss function is then calculated as the difference between the encoded input and the reconstructed output. Wehave thus built the main suggestive GAN model (as shown in Figure 8) by making two convolutional networks, discriminator and generator. Discriminator is build on the sequential model type of Keras whereas Generator involves linkage to custom layers [19].

While training, the Generator and Discriminator are coupled together. The generator output is attached to the discriminator input as in Figure 9. During every iteration, the discriminator is trained first using two data-the output of generator with fake label and labelled samples from the dataset. The parameters of the discriminator are then made static. Now the input of discriminator i.e. noise and latent code is forced to real labels enabling the training of only generator parameters. The latent code here is derived from the original dataset and mapped to the output with high mutual information.

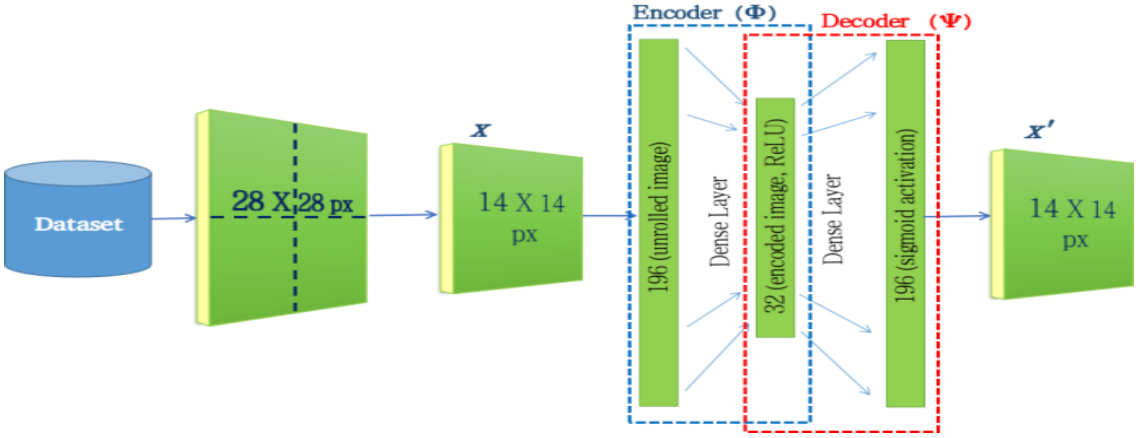



Figure 6. Autoencoder structure and image data flow diagram

```
Layer (type)                 Output Shape
=================================================
input_1 (InputLayer)         (None, 196)
_________________________________________________
dense_1 (Dense, ReLU)        (None, 32)
_________________________________________________
dense_2 (Dense, sigmoid)     (None, 196)
=================================================
Total params: 12,772
Trainable params: 12,772
Non-trainable params: 0
```

Figure 7. The Network architecture of the autoencoder

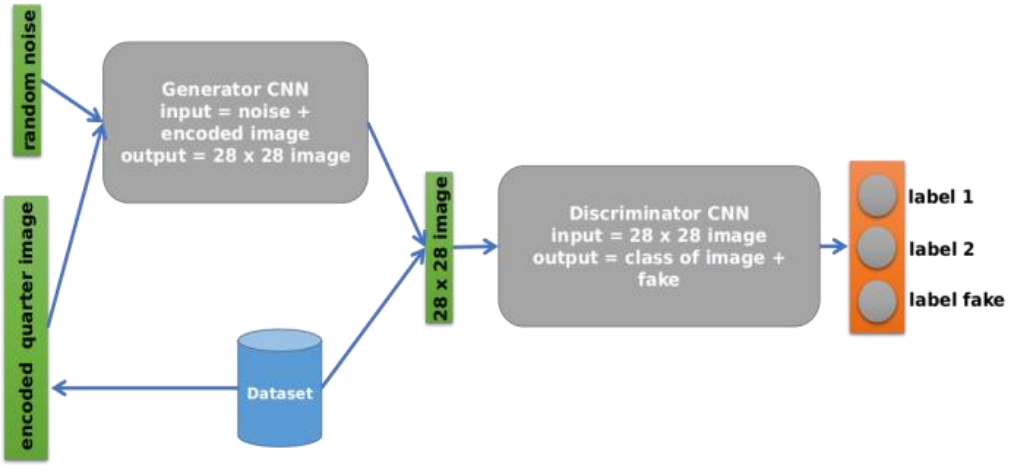



Figure 8. Suggestive GAN architecture. The Generator and discriminator coupled together during the training process. The dataset is used to derive the encoded quarter which is served as latent input to the Generator. The output of the discriminator - labels and fake

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| label_input (InputLayer) | (None, 32) | 0 | |
| latent_input (InputLayer) | (None, 100) | 0 | |
| concatenate_1 (Concatenate) | (None, 132) | 0 | label_input[0][0]<br>latent_input[0][0] |
| dense_1 (Dense) | (None, 6272) | 834176 | concatenate_1[0][0] |
| leaky_re_lu_1 (LeakyReLU) | (None, 6272) | 0 | dense_1[0][0] |
| reshape_1 (Reshape) | (None, 7, 7, 128) | 0 | leaky_re_lu_1[0][0] |
| up_sampling2d_1 (UpSampling2D) | (None, 14, 14, 128) | 0 | reshape_1[0][0] |
| conv2d_1 (Conv2D) | (None, 14, 14, 64) | 73792 | up_sampling2d_1[0][0] |
| leaky_re_lu_2 (LeakyReLU) | (None, 14, 14, 64) | 0 | conv2d_1[0][0] |
| batch_normalization_1 (BatchNor | (None, 14, 14, 64) | 256 | leaky_re_lu_2[0][0] |
| up_sampling2d_2 (UpSampling2D) | (None, 28, 28, 64) | 0 | batch_normalization_1[0][0] |
| conv2d_2 (Conv2D) | (None, 28, 28, 32) | 18464 | up_sampling2d_2[0][0] |
| leaky_re_lu_3 (LeakyReLU) | (None, 28, 28, 32) | 0 | conv2d_2[0][0] |
| batch_normalization_2 (BatchNor | (None, 28, 28, 32) | 128 | leaky_re_lu_3[0][0] |
| conv2d_3 (Conv2D) | (None, 28, 28, 1) | 289 | batch_normalization_2[0][0] |
| activation_1 (Activation) | (None, 28, 28, 1) | 0 | conv2d_3[0][0] |
| reshape_2 (Reshape) | (None, 28, 28) | 0 | activation_1[0][0] |

(a)

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, 28, 28) | 0 |
| reshape_3 (Reshape) | (None, 28, 28, 1) | 0 |
| conv2d_4 (Conv2D) | (None, 14, 14, 16) | 160 |
| leaky_re_lu_4 (LeakyReLU) | (None, 14, 14, 16) | 0 |
| batch_normalization_3 (Batch | (None, 14, 14, 16) | 64 |
| conv2d_5 (Conv2D) | (None, 7, 7, 32) | 4640 |
| leaky_re_lu_5 (LeakyReLU) | (None, 7, 7, 32) | 0 |
| batch_normalization_4 (Batch | (None, 7, 7, 32) | 128 |
| conv2d_6 (Conv2D) | (None, 4, 4, 64) | 18496 |
| leaky_re_lu_6 (LeakyReLU) | (None, 4, 4, 64) | 0 |
| batch_normalization_5 (Batch | (None, 4, 4, 64) | 256 |
| conv2d_7 (Conv2D) | (None, 2, 2, 128) | 73856 |
| leaky_re_lu_7 (LeakyReLU) | (None, 2, 2, 128) | 0 |
| dropout_1 (Dropout) | (None, 2, 2, 128) | 0 |
| batch_normalization_6 (Batch | (None, 2, 2, 128) | 512 |
| flatten_1 (Flatten) | (None, 512) | 0 |
| dense_2 (Dense) | (None, 3) | 1539 |

(b)

Figure 9. Generator network. (a) Generator accepts noise and encoded latent as input and generates a 28x28 output. There are various dense and convolution layers with up-sampling, batch normalization and Leaky ReLU activation function. (b) Discriminator network takes 28x28 image data and maps to the image label or fake label. This is constructed on the Sequential format of Keras i.e. each layer is followed by precedes another layer except the input and output layer.

The complete process discussed above is described in the flowchart Figure 10. We train the autoencoder for the entire dataset once before starting epochs and for a random set of n images selected from the dataset, we predict the latent input with trained model. All the epoch count, batch size and encoding dimension are subject to dataset image structure, differentiation and experimentation. Since the structural differentiation of the two classes i.e. apple and pencil are substantial, the epochs and latent input size can be lower but as we span across multiple class images or similar images, we need to keep these two parameters sufficiently high.

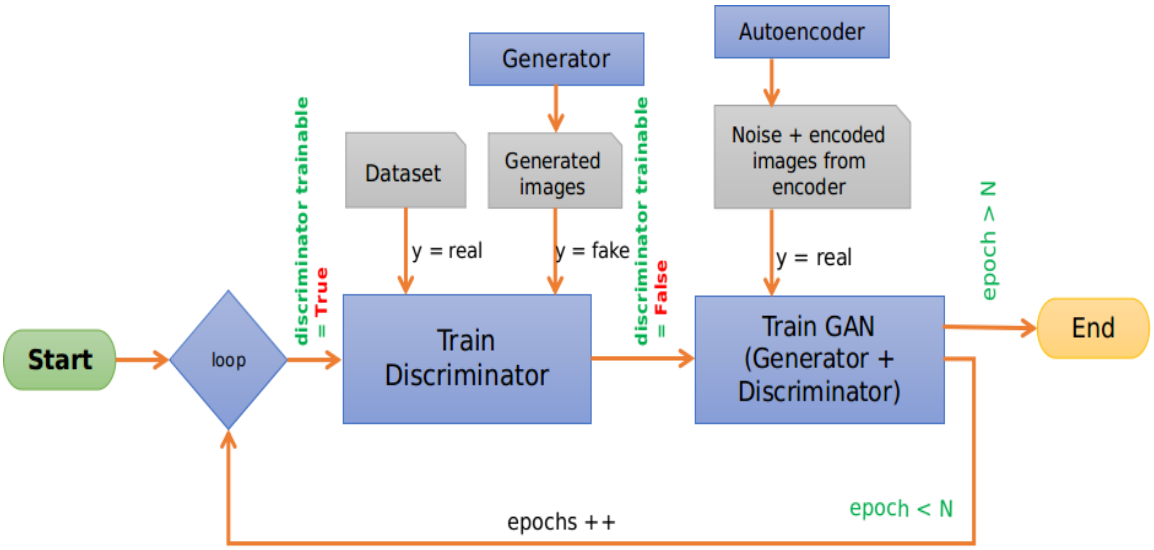



Figure 10. Flow chart of the Suggestive GAN training process

The latent input of size 2 just enables differentiation between the two classes i.e. for class 1 the input to the latent space is observed as [ 0.77, 0.14] and for class 2 [0.21, 0.71] for 2 random samples. Most of the details of the images structure fades away subsequently creating similar images for one class as shown in Figure 11(a). This property is observed for the images as shown in Figure 11(b). Every pencil distinguished is upright and every apple is uniformly round neglecting every structural property of the quarter image. The differentiation with respect to position and shape increases and we can observe that enough details is acquired for encoding dimension 16. For 32 latent spaces, we get the precise output as shown in Figure 11(c). Thus, the SGAN model has successfully rendered the final sample images of apple and pencil as was originally present in the Google draw dataset even when the input was partially provided to the model. Here's a comparision of some other significant research regarding reconstruction of images from partial occlusion. Let us now have a look at the output of discriminator under different latent input condition as shown in Figure 12.

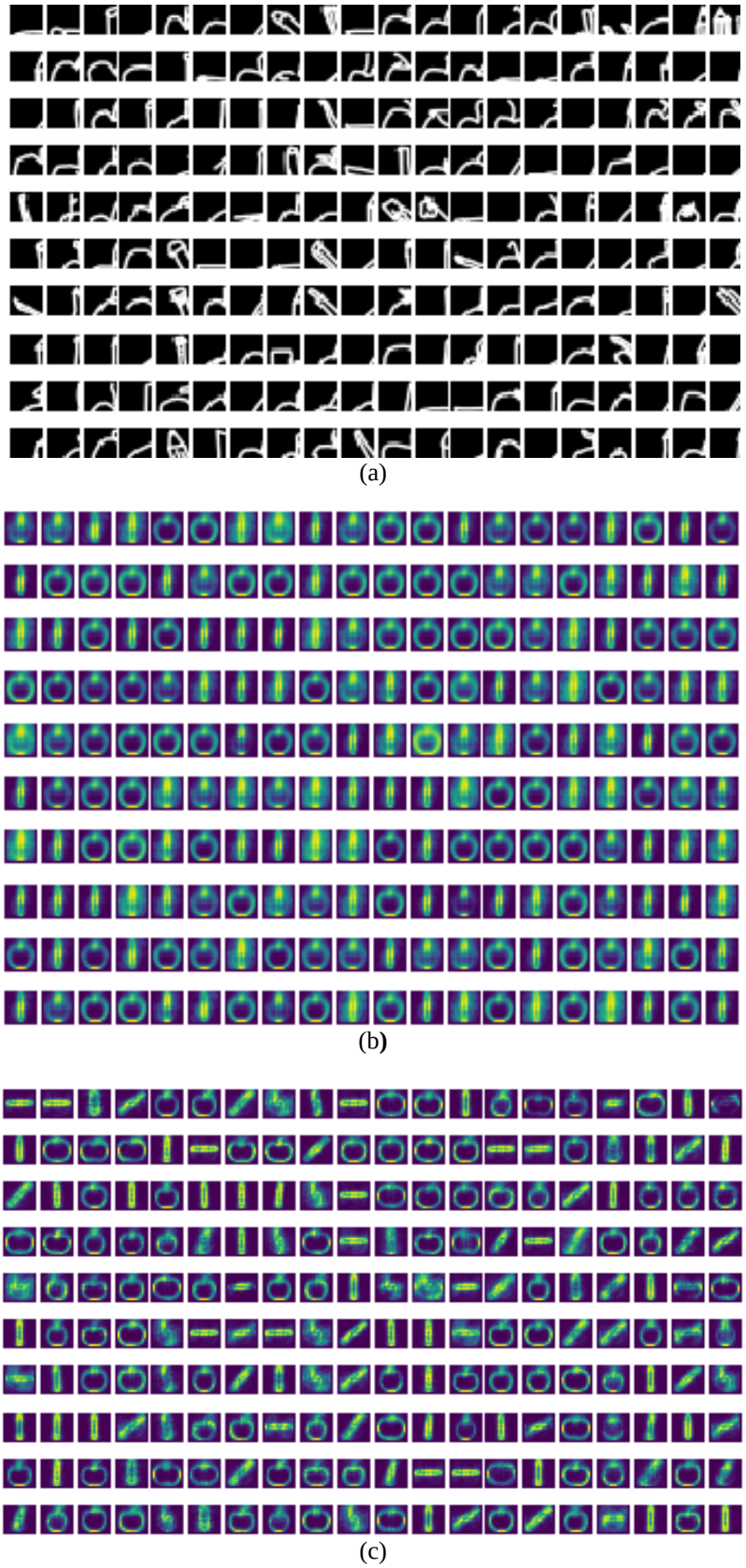


Figure 11. (a) The clipped images which when encoded serves as input, (b) The generated images when the encoding dimension is 2, c) The generated images when encoding dimension is 32

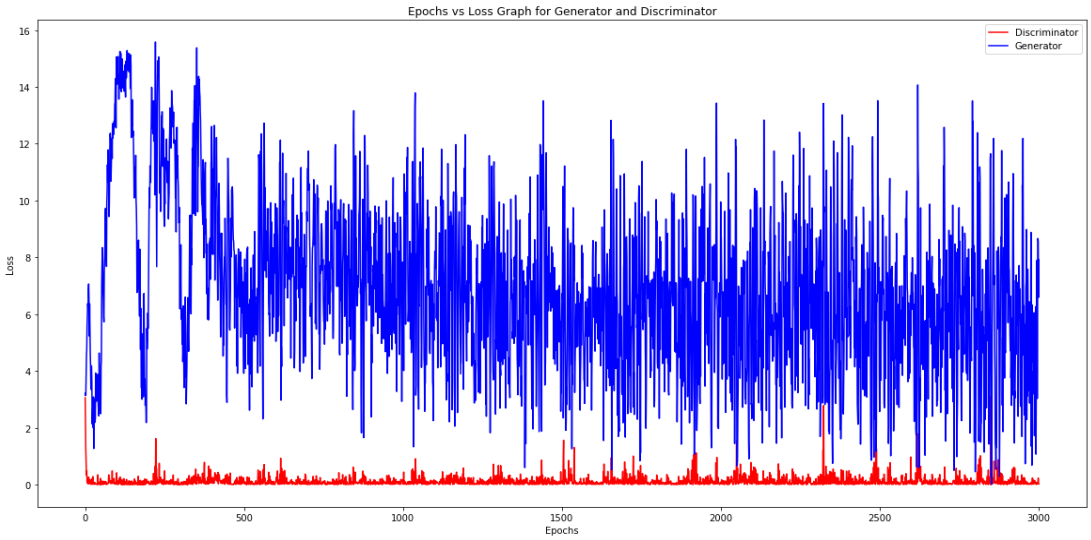



Figure 12. The Discriminator and Generator loss with respect to the number of epochs graph

## 6. CONCLUSION AND CONTRIBUTION

The major recognition phenomenon as proven and proposed by the SGAN model is that as the size of the latent space is increased, the images generated start inheriting the structural information received from the input image. The algorithm learns over the semantics such that most of the details of the images structure fades away, subsequently creating similar images for one class. This property is observed for the real drawings taken as sample by the special abled dysgraphic children and also validated on standard dataset images as shown in fig 9(b). It was thus observed that every image of a pencil distinguished is upright and every apple is uniformly round neglecting every structural property of the quarter image. This feature is very useful when we have distorted images or imagery with missing strokes too. Our proposed SGAN model reconstructs the complete image and recognises the initial object in the latent space with utmost accuracy. Future contribution would be to improvise this model to voluminous unstructured image data and assist the Learners Management System.

## REFERENCES

[1] Mescheder, L., Nowozin, S., Geiger, A.: Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. arXiv preprint arXiv:1701.04722 (2017).
[2] Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, in: International Conference on Learning Representations (ICLR), arXiv:1511.05644, San Juan, 2016.
[3] Ming-Yu Liu and Oncel Tuzel, "Coupled Generative Adversarial Networks" in NIPS 2016.
[4] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. arXiv preprint arXiv:1612.04357, 2016.
[5] Che, Tong, Li, Yanran, Jacob, Athul Paul, Bengio, Yoshua, and Li, Wenjie. Mode regularized generative adversarial networks.arXiv preprint arXiv:1612.02136, 2016.
[6] Xiaoguang Han, Chang Gao, and Yizhou Yu. 2017. DeepSketch2Face: a deep learning based sketching system for 3D face and caricature modeling. ACM Trans. Graph. 36, 4, Article 126 (July 2017), 12 pages. DOI: https://doi.org/10.1145/3072959.3073629.
[7] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 151-160. DOI: https://doi.org/10.1145/1449715.1449740.
[8] Song, Y., Zhang, Z., & Qi, H. (2017). Recursive Cross-Domain Face/Sketch Generation from Limited Facial Parts. CoRR, abs/1706.00556.
[9] Cherlin, J.J., Samavati, F.F., Sousa, M.C., & Jorge, J.A. (2005). Sketch-based modeling with few strokes. SCCG.
[10] Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NIPS, 2014, pp. 2672–2680.
[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression", arXiv:1804.09535v1 [cs.CV], 2018.
[12] Thierry Dumas, Aline Roumy, Christine Guillemot. Autoencoder based image compression: can the learning be quantization independent?. ICASSP 2018 - International Conference on Acoustics, Speech and Signal Processing

ICASSP, Apr 2018, Calgary, Canada. IEEE, pp.1188-1192, 2018, <10.1109/ICASSP.2018.8462263>. <hal-01713644>
[13] J. Almotiri, K. Elleithy and A. Elleithy, "Comparison of autoencoder and Principal Component Analysis followed by neural network for e-learning using handwritten recognition," 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, 2017, pp. 1-5.
[14] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets" eprint arXiv:1606.03657, 2016.
[15] Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," ArXiv preprint arXiv:1511.06434, 2015.
[16] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. J. Mach. Learn. Res. 11 (December 2010), 3371-3408.
[17] D. Barber and F. V. Agakov, "Kernelized infomax clustering," in NIPS, 2005, pp. 17–24.
[18] Sento, "Image compression with auto-encoder algorithm using deep neural network (DNN)," 2016 Management and Innovation Technology International Conference (MITicon), Bang-San, 2016, pp. MIT-99-MIT-103.
[19] D. Barber and F. V. Agakov, "The IM algorithm: A variational approach to information maximization," in NIPS, 2003.
[20] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. ACM Trans. Graph. 33, 4, Article 129 (July 2014), 10 pages. DOI: https://doi.org/10.1145/2601097.2601205.
[21] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. ACM Trans. Graph. 36, 4, Article 107 (July 2017), 14 pages. DOI: https://doi.org/10.1145/3072959.3073659
[22] J. Gautier, O. Le Meur and C. Guillemot, "Depth-based image completion for view synthesis," 2011 3DTV Conference: The TrueVision - Capture, Transmission and Display of 3D Video (3DTV-CON), Antalya, 2011, pp. 1-4.doi: 10.1109/3DTV.2011.5877193.

## BIOGRAPHIES OF AUTHORS

Smita Pallavi has done B.E. and PhD in Information Technology. She has teaching experience of 13 years and is handling major Govt. projects of GoI and GoB. She has published SCI and indexed Research journals in areas of Soft Computing, Data Mining and Optimization algorithms.

Akash Kumar has done B.E. from Birla Institute of Technology, Mesra Patna Campus in Information Technology branch. A Data Science enthusiast, his areas of interest include Machine learning, Deep Learning, Data Mining, and Computer Vision. With expertise in Python, he is also a regionallevel winner in Open Government Data Hack-2017 organized by NIC and Startup India and has several certifications from IEEE. He is an intern at SumyagData Sciences Pvt. Ltd. and has been employed by Lowe's India.

Abhinav Ankur is an Analyst in Data Science at Verizon Data Services India Private Limited and has professional experience in Data Engineering and Data Science. He has specialized in Churn Modeling, Real-Time Anomaly Detection, and creation & deployment of Interpretable Machine Learning Models using LIME, SHAP etc He is working towards implementation of enterprise "System of Insights" as of now and has won several hackathons conducted globally at Verizon.